AFRL-IF-RS-TR-2006-296
In-House Final Technical Report
October 2006

ESTABLISHING TOOLS FOR COMPUTING HYBRIDS

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

STINFO FINAL REPORT

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report was cleared for public release by the Air Force Research Laboratory Rome Research Site Public Affairs Office and is available to the general public, including foreign nationals. Copies may be obtained from the Defense Technical Information Center (DTIC) (http://www.dtic.mil).

AFRL-IF-RS-TR-2006-296 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE DIRECTOR:

/s/	/s/
LOIS D. WALSH, Chief Advanced Computing Technical Branch Advanced Computing Division	JAMES A. COLLINS, Deputy Chief Advanced Computing Division Information Directorate

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
OCT 2006	In-House Final	Oct 03 – Dec 05

4. TITLE AND SUBTITLE

ESTABLISHING TOOLS FOR COMPUTING HYBRIDS

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Clare Thiem, Sunil Bhat and Thomas Blount

5d. PROJECT NUMBER
459T

5e. TASK NUMBER
ET

5f. WORK UNIT NUMBER
CH

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

AFRL/IFTC
525 Brooks Rd
Rome NY 13441-4505

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

AFRL/IFTC
525 Brooks Rd
Rome NY 13441-4505

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSORING/MONITORING AGENCY REPORT NUMBER
AFRL-IF-RS-TR-2006-296

12. DISTRIBUTION AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED. PA# 06-685

13. SUPPLEMENTARY NOTES

14. ABSTRACT
Establishing Tools for Computing hybrids (ETCH) provides a starting point from which to explore how advanced computing technology, based upon biotechnology, might enable intelligent systems and provide new capabilities to the Air Force. Much effort was spent exploring published research to understand what others have accomplished, problems they encountered, the potential of the biotechnology to advance computing technology, and identifying potential paths on which to focus future research. Various computational biology tools were also examined and found to have the potential to help develop biotechnology for future computing architectures. This project provided the opportunity to gain hands on experience to complement the knowledge gained from participation in various DARPA Programs that involved biotechnology and related modeling and simulation capabilities. It established a foundation on which new research projects can be built to take biotechnology from the laboratory into the field as advanced hybrid information systems.

15. SUBJECT TERMS
Biomolecular Computing, Bioinformatics, Computer, Computational Biology

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Clare Thiem
U	U		U	39	19b. TELEPHONE NUMBER (*Include area code*)

Table of Contents

List of Figures

1.0 Introduction

Establishing Tools for Computing Hybrids (ETCH) provides a starting point from which to explore how advanced computing technology might enable intelligent systems and provide new capabilities to the Air Force. One research thrust of the Advanced Computing Architectures Focus Area within the Air Force Research Laboratory's Information Directorate (IF) examines novel information processing paradigms for new, radical methods of processing information.[53] This research encompasses nanotechnology, biotechnology and the quantum sciences. The scientists and engineers conducting the research are looking for the means to either conduct existing computational tasks better or address computational problems that classical computing has difficulty solving. Classical computing for this project was viewed as silicon based computing utilized in general purpose computing. Examples of computing that would be referred to as non-classical are quantum and biomolecular computing. Within the Advanced Computing Technology Branch (IFTC) of the Information Directorate the interest in bio-molecular computing and quantum computing is pursued under the perspective that biomolecular and quantum computing do not represent technologies that will replace general-purpose silicon computing, but represent complementary technologies. The common viewpoint is that as these technologies mature they will be realized through the creation of a hybrid architecture that integrates a combination of biotechnology, quantum science, nanotechnology, and general purpose silicon-based computing technology.

Over the past several years, members of the Information Technology Division have supported several Defense Advanced Research Projects Agency (DARPA) Programs that fall under the broad category of biotechnology. These programs were: Bio-Fluidic Chips (BioFlips) that demonstrated the integration of biofluidics with electronics, Simulation of Biological Systems (SIMBIOSYS) which expanded the capability of multiphysics design tools, and the Bio-Computation (BioCOMP) Program which sought to, "...explore, develop and exploit both the role computation plays in biology and the role biology plays in computation."[25] Unfortunately, the role biology plays in computation; a primary interested for the IF's involvement in the BioCOMP Program was terminated early due to budget cuts. A majority of the technology demonstrations for these programs focused on the health and medical fields since they hold the strongest promise of future funding to develop the technology further. The results of these programs will be leveraged in demonstrating how biotechnology can provide new capabilities for information dominance. Before the capability of a biologically-based information system concept can be demonstrated, however, more research is required to develop the technology to the level of maturity commensurate with practical application. This project examined the following three questions:

a. Can IF leverage biotechnology for information dominance?
b. If so, can computational biology tools such as those incorporated into Bio-SPICE be used by engineers to develop biotechnology to the point of maturity that would advocate integration into future information systems?

c. Can Bio-SPICE in particular be used to unleash the potential of biotechnology for information systems?

For the purposes of this report, the following definitions were used to categorize reviewed research:

Biomolecular Computing is computing with biological material.

Bio-inspired computing is the mimicking of biological processes on silicon.

Bioinformatics is, "an interdisciplinary field bringing together biology, computer science, mathematics, statistics, and information theory to analyze biological data for interpretation and prediction."[8] Basically, this involves the application of computer science to process data for medical and biological research.

Biocomputing is the broad term that covers the pursuit of computing technology which encompasses biomolecular computing, bio-inspired computing, and bioinformatics.

This report begins by examining how biotechnology can be used in future advanced information systems. It then examines computational biology tools, such as those found in the Bio-SPICE open source software environment, which may be used to develop biotechnology. The report then discusses how Bio-SPICE could be used to develop the biotechnology for information systems examining ideas for biomolecular computing. Finally, technology gaps related to the development of hybrid computing architectures will be presented before concluding. Prior to this effort, AFRL/IFTC had no in-house experience with computational biology tools and a limited knowledge of this class tools was obtained from managing efforts funded by DARPA.

2.0 Biotechnology for Advanced Information Systems

In order to better understand how the Information Directorate could benefit from biotechnology, an extensive literature search was conducted. Discussions were also held with various researchers during different DARPA Principal Investigator Meetings, conferences, site meetings, etc. to determine the future direction of biotechnology. Articles and books were organized by potential application of biotechnology. The main application categories are: computing, memory, network security/management, nanofabrication, self-assembly/reconfigurability, and bioinformatics.

2.1 Computing

Computing is one application area for biotechnology that is of particular interest for the Advanced Computing Architectures Focus Area at the Information Directorate. Biotechnology holds the promise of addressing problems that have proven difficult or even possible to address with classical computing. "The present interest in biocomputing

is due to many factors. The first impetus arises from electronics. Limits are extended year after year, yet as some point the size, speed, and power dissipation of switches based on silicon or other conventional materials will run into the deadlocks set by the basic laws of physics. Already the 'leakiness' of quantum effects between nano-sized wires that are nanometers apart has sparked interest in somehow harnessing the power (quantum cellular automata, etc.) of quantum mechanical devices. The second impetus is that although conventional computer science has been extremely successful, a number of critical problems in information processing have persisted stubbornly beyond reach: pattern recognition, learning, and parallelism being three examples of where biological systems still remain far advanced beyond their silicon mimics."[48] Several articles have looked at the physical limits of current computer technology [14, 50, 58] and debated whether or not the demise of Moore's Law is on the horizon necessitating the need for alternative computing technologies [24, 28, 30, 34, 42, 46, 52, 55, 62].

There are many different theoretical perspectives on how one might leverage biotechnology to achieve new computing capability which makes it difficult to locate the best path to a practical, realizable concept. Highlighting the diversity is the number of names of biocomputing that were encountered during the literature review which included: DNA computing, molecular computing, biomolecular computing, bioinspired computing, membrane computing, reaction-diffusion computing, and chemical based computing. Periodically papers were written that provided a snapshot of what progress was made, the direction of ongoing research, and research challenges. [10, 15, 39, 40, 41] The literature review reaffirmed, that it is not a question if biotechnology can be leveraged and developed into new technologies, but when biotechnology will be integrated into advanced systems. One source supporting this opinion is The International Technology Roadmap for Semiconductors (ITRS) which includes information regarding molecular devices, molecular memories and biologically inspired architectures as experts look at the status of the semiconductor industry and its future direction. The roadmap perspective is that of emulating human and biological reasoning functions through biologically inspired computing which utilizes the technology growth of the semiconductor industry and the existing infrastructure. Metrics were established on which to measure the progress. One set of metrics compiled for the ITRS are the fundamental parameters of the human brain [22] which include:

a. Number of neurons – 2E10
b. A single neuron can make 100 to 10,000 synaptic connections
c. Power consumption – 15-30 Watts
d. Information Stored – 1E12 (short term) bits
e. Information process speed – 1E13 bits/second

A question that came to mind, as the ITRS information was reviewed, is if we restrict ourselves to developing only biotechnology supported by the infrastructure of the semiconductor industry, are we inhibiting the exploitation of the full power of biology for computing? At this time, the answer is yes because we are far from understanding the true potential of biotechnology. Even the ITRS, with its narrowed focus, acknowledges that much more research is still needed. Garzon and Deaton stated that, "It is increasingly

clear, however, that understanding the actual power of biomolecules to solve computational problems in practice requires developing the notion of complexity that captures the physico-chemical reality in which they take place (entirely different from VLSI-based programs), so that it results can be used to gauge the true computational power of molecular computing."[15] They also discussed the fundamental challenges that needed, "to be resolved for bringing molecular computing to an effective new paradigm for computation science."[15] The challenges are:

a. Reliability, efficiency and scalability
b. The encoding problem
c. Error-preventing codes
d. Building and programming molecular computers
e. Implementing evolutionary computation
f. Autonomy and self-assembly
g. Molecular computability and complexity

A list of resources for interested parties to obtain information on the progress of the research is provided in their paper. The concepts of DNA Computing, Immunocomputing, Membrane Computing and Cellular Computing will be discussed in more detail below.

DNA computing involves the utilization of a fundamental biological building block to carry out computational tasks. DNA is an example of an information-carrying polymer. "A number of labs are taking different approaches, but all take advantage of the complementarity of the two strands of double-stranded DNA and the ability of strings of DNA subunits (Called bases or nucleotides) to bind together using Watson-Crick pairing (A[adenosine] with T[thymidine] and C[cytosine] with G[guanidine]), says Seeman."[27] Adleman demonstrated DNA computing in 1994 on a Hamiltonian Path Problem.[1] While the approach he used is not practical for large problems his experiment demonstrated two things. First, that computing can be done with biological material in a controlled environment. Second, computing can be conducted with very small structures, structures much smaller than the current state-of-the-art computing technology. His success also raised the interest of other researchers sparking renewed interest in leveraging biology in order to obtain new computational capability. McCaskill, van Noort, and others continue to pursue DNA computing developing more practical means of implementing the computing concept.[32, 33, 56, 57] Some research teams looked at the DNA strand design issue for classical computation and self-assembly computation [6], others examined the programming environment for DNA computing[7], while others addressed the role surface science will play in DNA computing becoming a reality.[16] One overview of this area of research stated: "Which tools and algorithms will turn out best for DNA computing is still an open question." [29] DNA, however, is not the only aspect of biology researchers are pursuing.

"The natural immune system is a subject of great interest because it provides an excellent model of adaptive processes operating at a local level and of useful behavior emerging at the global level; therefore, it inspires new powerful paradigms for information processing

and computing."[63] The immediate application of artificial immune systems, a bio-inspired technology, is network security which will be discussed in more detail later. It also seems that, after a brief review of published work, the functionality of the immune system could play a role in the development of cognizant computing or in the processing of massive amounts intelligence gathered from different sources. Researchers at the Jet Propulsion Laboratory looked at a means to incorporate the, "...ability to take the position of an observer in relation to one's own thoughts..."[63] While researchers struggle to decipher and understand the natural immune system, several concepts repeatedly surfaced such as the immune systems ability to recognize patterns, detect abnormal behavior, and respond to ever changing threats. Maturity of this technology area continues as researchers gain more knowledge just like other bio-inspired technologies such as membrane computing.

The concept of membrane computing is relatively new, coming into existence in 1998 when Gheorghe Paun published his first paper on the subject. He stated that, "It must be stressed that membrane systems (also called P systems) are not intended to model the functioning of biological membranes."[35] "Membrane Computing is a bio-inspired branch of natural computing, abstracting computing models from the structure and functioning of living cells and from the organization of cells in tissues or other high order structures." [37] "In short, it deals with distributed and parallel computing models, processing multisets of symbol-objects in a localized manner (evolution rules and evolving objects are encapsulated into compartments by membranes), with an essential role played by the communication among compartments (with the environment as well)...The essential ingredient of a P system is its member structure, which can be a hierarchical arrangement of membranes, like a cell (hence described by a tree), or a net of membranes (placed in the nodes of a graph), like in a tissue, or in a neural net."[36] Figure 1 illustrates the concept of a classical P system that Paun has used in several papers. Figure 2 shows the tree diagram of the same example.

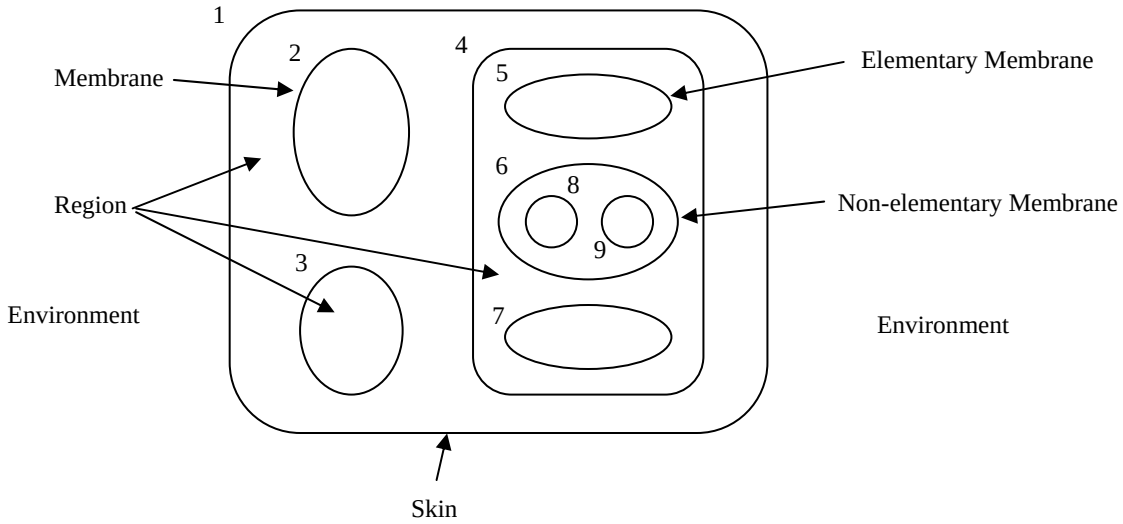


Figure 1. Illustration of a membrane structure of a P system with fundamental components highlighted.

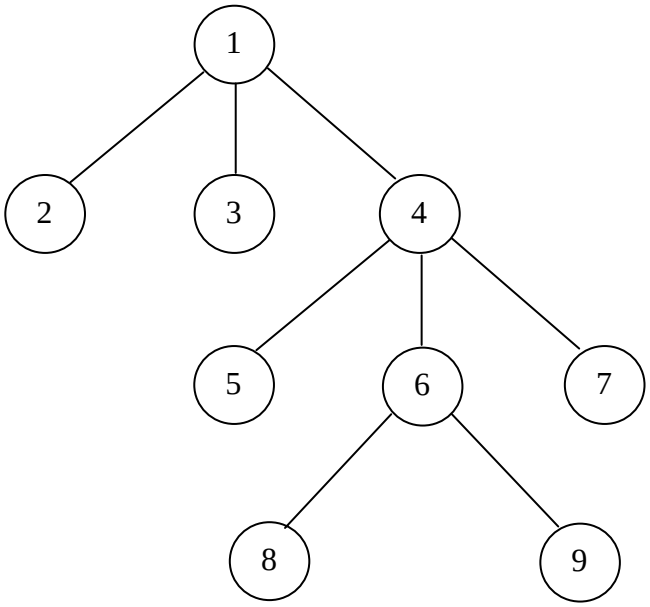


Figure 2. Tree diagram for example membrane structure sample illustrated in Figure 1.

Several variations of P systems exist. A good web site for additional information is: http://www.psystems.disco.unimb.it. Features of P systems that make it attractive for different applications are: distribution, discrete mathematics, algorithmicity, scalability/extensibility, transparency, parallelism, non-determinism and communication.[36] Application areas include: systems biology, economics, computer graphics, linguistics, and management. The concept of actually computing with cells is a different category of computing.

Cellular computing is, "A computing philosophy based on simplicity, vast parallelism and locality."[47] Several different definitions of cellular computing were found during the literature review. Some researchers define it as computing with living cells while others refer to it as computing with engineered cells or synthetic cells. "Sipper defines three principles of cellular computing:

1. Simplicity: The processor in cellular computing is defined as a cell. It can do very little on its own. The system performs complex tasks through the combined function and cooperation of many cells.
2. Vast Parallelism: Most parallel computers have at most 40-60 processors. 'Massively parallel' computers, as they are called, contain thousands or tens of thousands of processors. In this system, the parallelism is exponential, with 10X processors.
3. Locality: Each cell communicates only with other cells that are close by, and each communication contains only a small amount of information. There is no

single cell that has an overview of the computer system, and no single cell that controls the entire system i.e. there is no CPU or central processing unit.

Systems which use bacteria as substrates for engineering meet these three criteria for a cellular computing system."[47] The key to this computing concept is controlling the behavior of cells. It was recognized that the processing speed of hardware will be slow, but it may be possible to compensate for this deficiency with the appropriate software. Research in Amorphus Computing has been identified as providing key insight into programming approaches appropriate for cellular computing. [2, 47]

While numerous researchers looked at portions of the technology needed to realize cellular computing some of the more significant research leading to practical implementation of cellular computing is at the Massachusetts Institute of Technology and Princeton. This research provides a basis for IF/IFTA to pursue collaboration with researchers in the Human Effectiveness Directorate. Weiss et al. published progress made in developing technology for cellular computing, or more precisely, "creating synthetic gene networks for modifying and extending the behavior of living organisms." [60]. "The main challenge in biocircuit design lies in selecting well matched genetic components that when coupled, reliably produce the desired behavior." [60] Research Weiss was involved with developed a design process to allow modular, "construction of complex circuits using a library of interchangeable components." [61] It was stated that, "While the field holds great promise, we still face a number of challenges. One of the major obstacles is our current inability to devise models and perform simulations that can accurately predict the quantitative behavior of genetic networks. Thus, the difficulty of inferring function from DNA sequence limits our circuit design efficiency. In addition, we must delineate the limits of the new information processing capabilities that can be embedded into cells. Beyond the engineering of individual cells, new programming paradigms are needed to achieve the coordinated behavior of cell aggregates. Biological substrates are constrained by factors such as unreliable computing elements, significant noise, and imperfect communication with limited range. Therefore, our designs must strive to achieve sufficient reliability and reproducibility. Such a myriad of challenges leave much opportunity for future work." [60] In another article it was pointed out that, "We do not have a complete list of all repressors and the sites to which they bind. We do not have accurate or complete data on the kinetic constants involved. We still do not understand the metabolism or reproduction of cells well enough to accurately predict the effects of any interference with them." [47]

As the cellular computing concept and technology mature they, "...will enable cells to perform sophisticated digital and analog computation, both as individual entities and as part of larger cell communities. This engineering discipline and its associated tools will advance the capabilities of genetic engineering, and allow us to harness cells for a myriad of applications not previously achievable."[60] Other application areas include: human therapeutics, synthesis of pharmaceutical products, molecular fabrication of biomaterials, nanomachine assembly, and toxin detection with biological sentinels.

2.2 Memory

Biotechnology may also provide a means to store and retrieve information more efficiently. The application of biotechnology to advanced memory technologies represents the most likely path to the first practical realization of the technology in an information system. There are many different proposals for molecular memory recognized in the ITRS. One briefly explained example involved the storage of data that functioned, "by applying an external voltage that causes the transition of the molecule into one of two possible conduction states. Data is reading by measuring resistance changes in the molecular cell." They went on to state that, "There are also concepts for combining molecular components with current memory technology, such as DRAM and floating gate memory."[22] In the research legacy associated with the Information Directorate, there was an investment in three dimensional protein-based optical memory utilizing bacteriorhodopsin and a sequential one photon process. Spanning several efforts, the process was refined, material optimized, and prototypes built striving to provide the foundation for a commercially viable optical memory. [3,4,5] Two other approaches to protein-based volumetric memories are holographic and simultaneous two-photon. [31] The Australian Broadcasting Company recently reported that a prototype device was produced setting the stage for a bacteriorhodopsin coated DVD projected to hold 50 terabytes of information. It is estimated that a USB disk will be commercialized by July 2007 and a DVD version by July 2008. [44]

2.3 Network Management and Security

Network management and security represents an area that will benefit from bio-inspired technology. Researchers are exploring how the natural immune system responds to abnormal events or foreign invaders and how this knowledge can be applied to computers and computer networks. At the time of this literature review the impression was that, for this application of biotechnology, research is focused on replicating particular end states in phases of immune system functionality, not the biological processes that led to the end state. This is primarily due to the fact that, "...the biological machinery of the immune systems is poorly understood, the only alternative is to mimic the phenomenology of its performance using some equivalent physical models." [62] Using the natural immune system as a guide, the computer security research teams are trying to develop analogous techniques for detecting abnormal behavior on computers or computer networks, and determining the proper response. This research can be referred to as artificial immune systems, immunocomputing, or immunity-based techniques. Interest in this bio-inspired technology for computer technology has grown since 1994.[9] University of Memphis researchers, for instance, are interested in developing a technique which could differentiate, "various degrees of abnormality in network traffic.[9] Researchers at the Air Force Institute of Technology examined, "the use of a distributed-agent biological-system approach toward the computer-security problems of virus elimination and ID."[17] Florida Institute of Technology research has examined the use of an artificial immune system for securing ad hoc networks against intrusion attacks.[19] It was stated that, "As with biological immune systems, the problem of protecting computer systems from malicious intrusions can be viewed as the problem of distinguishing 'self' from

dangerous ‘other’ (or ‘non-self’) and eliminating this ‘other.’”[51] As the dependency on computers grows and knowledge of natural immune systems increases, this relatively new area of bio-inspired technology is likely to play a bigger role.

2.4 Nanofabrication/Self-assembly/Reconfigurability

Future nanoelectronic hardware will benefit from biotechnology used to fabricate new devices and architectures when current fabrication techniques reach their scaling limits. Traditional lithographic based fabrication approaches will be replaced by processes that leverage the natural ability of biological organisms to bind to a receptor protein, assemble multiprotein complexes, etc. Molecular recognition plays a major role in self-assembly and self-recognition. “Molecular recognition is the binding and specific selection of substrate(s) by a given receptor molecule....Self-assembly refers to the more or less automatic putting together of parts, based almost purely on molecular recognition. Self-organization refers to self-assembly with the generation of organized entities and may result in networks and feedback systems on different levels.”[49] “Molecular synthesis describes the fabrication of molecules and macromolecules for application to molecular devices ..., thin films, directed, and self-assembly agents. Improved understanding of the relationship between molecular structure and electronic properties may enable design and application of new molecules with unique properties as new elements of emerging research devices. Furthermore, needs for sub-nanometer directed and self-assembly may require molecules designed to recognize other molecules or structures as registration features. Thus, research is needed to understand the relationship between molecular structure and potential assembly mechanisms and their limitations in directed assembly.”[22] Researchers at Duke University, for instance, are not only conducting research on how to fabricate DNA self-assembled computer architectures, but are also developing tools and techniques to make the technology commercially viable. They looked at carbon nanotube transistors self-assembled into circuits using DNA [11] along with metallized DNA to form decoupled array multi-processors and oracles.[12] They also developed a system level design approach to evaluate self-assembled computer architectures.[13] Several agencies are funding research and IF has executed contractual efforts as it pursues the integration of biotechnology into future information systems.

2.5 Bioinformatics

Bioinformatics, as it pertains to the Information Directorate, can be looked at from a couple of different viewpoints. First, one can seek to exploit biological databases and other experimental results to leverage other researchers’ progress to facilitate the development of biotechnology for new applications. This is the viewpoint most likely supported by researchers interested in developing new computing capabilities. Second, it is an application area that can leverage AFRL/IF’s experience with databases, managing information, and fusing different sources of information for sound decision making. Experimentally, biologists and other researchers can gather data, but the problem is in understanding the information. There is a wide, “...range of applications of bioinformatics to molecular biology, clinical medicine, pharmacology, biotechnology, agriculture, forensic science, anthropology and other disciplines.” [26] Numerous computational

biology tools were developed utilizing a variety of bioinformatic approaches including statistical, modeling artificial neural networks, hidden Markov models, data mining, etc. Biologists and other scientists in various government organizations and other medical research communities have expressed their desire for AFRL/IF to invest in the second viewpoint. It has been stated that, "...many of the problems faced in bioinformatics are daunting in their size and scope."[8] Limited resources, however, constrains AFRL/IF's computer science expertise to focus primarily on command and control needs for the warfighter. The National Science Foundation provides funds for research and holds workshops to bring together bioinformatic researchers, users of molecular biology databases, and computer scientists to learn about what has been done and where to direct future funds.[23] Through programs such as BioCOMP, DARPA (with support from Information Directorate personnel) seeks to improve the role computation plays in biology.

3.0 Computational Biology Tools

Numerous independent computational tools are used by researchers to obtain a quantitative understanding of biology. The BioCOMP Program sought to address the non-integrated issue by developing an open-source environment in which researchers could bring together different software tools, exchange information by means of standard data formats, and access biological databases. Information Directorate involvement in the BioCOMP Program exposed the team to a community of researchers and their tools about which Information Directorate personnel previously had limited knowledge and no hands on experience. Directorate personnel do, however, have experience with computational tools from other disciplines including VLSI and microsystem design tools. The primary reason for supporting the BioCOMP Program was that part of the program was originally directed at understanding the role biology could play in computing. In order for technology to be realized, it will need to be supported with the appropriate modeling and simulation tools and techniques. Past experience indicates that a process for multiscale, multiphysics modeling and simulation is needed to bring new science to build future hybrid information systems. Figure 3 illustrates the different levels of multiscale, multiphysics modeling and simulation. These levels of simulation are based upon in-house microsystem design experience and are applicable to either a top-down or bottom up design approach. It was observed that designers with electronics engineering undergraduate degree are more likely to use a top down approach while those with a mechanical or aerospace engineering background were more comfortable with the bottom up approach. The three-dimensional (3-D) physical level represents the detailed analysis of the physical device likely involving finite element, finite difference, finite volume, or boundary element methods. The reduced order modeling level, commonly referred to as macromodeling in electronic design community, involves modeling of the control circuitry of the device with the performance characteristics of the device obtained in the 3-D physical modeling level and captured by a mathematical equation. The reduced order model represents a subsystem of the whole system in which the technology is to be integrated involving VHDL-AMS or SPICE modeling formats. The system level portrays the whole system bringing together all of the subassemblies utilizing VHDL, VHDL-

AMS, or VERILOG A formats. The biological level represents all of the new sciences that are being examined for exploitation in alternative computing concepts that will be used in hybrid computing architectures and not currently supported in common design tools and processes. The open question is how does one bring new science into the process? In the case of biotechnology this question leads to several more questions:

a. If one models a biological process at the molecular or cellular levels what information needs to be extracted from the simulation results in order to develop a practical computing concept?
b. What information must be passed between the different levels of modeling and simulation?
c. How is information passed between the different levels of modeling simulation?
d. Is the process a true bottom up design process, or can a level be skipped and the appropriate reduced order models be established without intensive three dimensional physical modeling?
e. Can computational biology tools be used to develop biotechnology for computing?
f. Can computational biology tools be integrated into a process similar to that used by microelectromechanical system designers?
g. Should computational biology tools be integrated into such a process?

These questions are difficult to address and it was never the intention of this effort to address them all, but to start building the foundation of knowledge that could begin the process. One of the first steps in building this foundation was to examine different computational biology tools and understand the type of problems they address. Participating in the BioCOMP Program provided an excellent opportunity to evaluate a variety of tools as it sought to produce an open source software environment for computational biology tools.

The portion of the BioCOMP Program that focused on improving the role computation plays in biology centered on the development of the Biological Simulation Program for Intra-Cellular Evaluation (Bio-SPICE). Bio-SPICE is intended to, "...assist in the clarification of the complexity inherent in the biochemical networks that operate within living cells. Intracellular biochemical networks work in a highly orchestrated fashion to regulate and control cell behavior, and often employ complex positive and negative feedback loops that can introduce strong nonlinearities into the system. The dynamics of such systems, even in relatively simplified mathematical representations, can easily surpass the reasoning capabilities of the unaided human mind. Helping the life sciences community begin to unravel this complexity, BioSPICE aims to provide a suite of software tools and reusable model elements that augment the capabilities of both 'belly to the bench' and 'theoretical' biologists with powerful mathematical and automated reasoning approaches. More specifically, the system has been designed to accommodate both naive and the power users." [25] As the program evolved, the various teams changed along with the Bio-SPICE software environment and the computational tools that worked

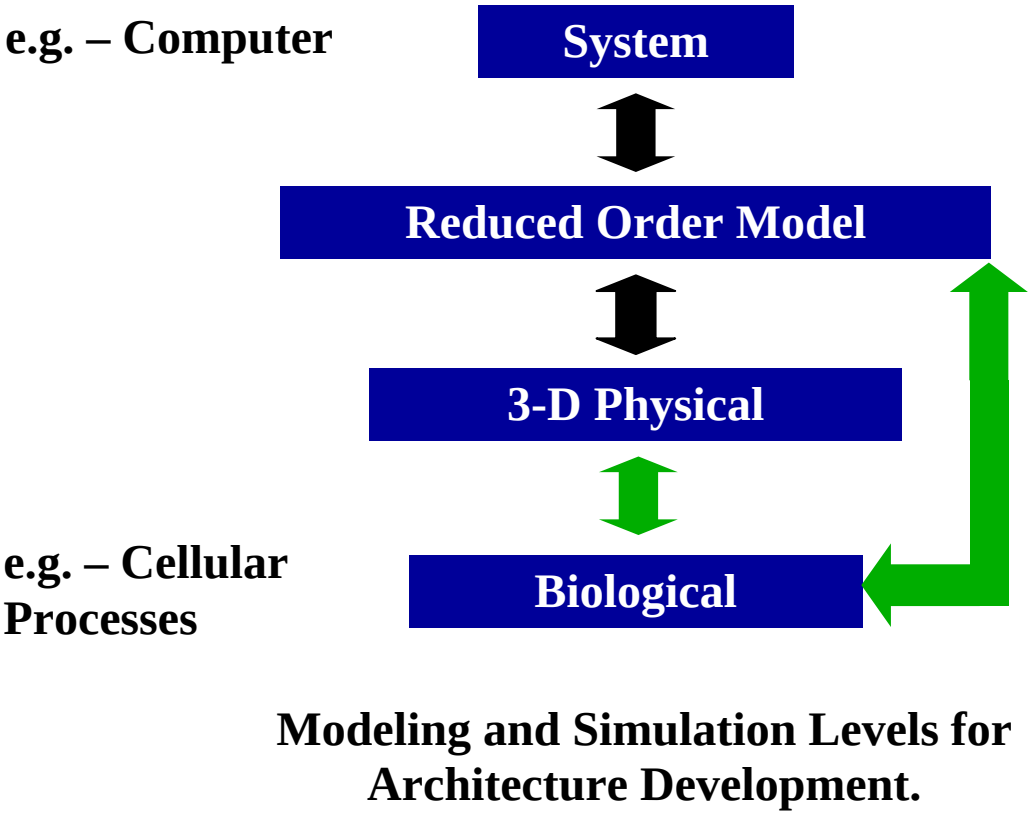


Figure 3. Illustration of the various levels of modeling and simulation required for developing hybrid architecture.

in it. Appendix A provides a snapshot of different tools available in the BioCOMP Program. The information in the table was extracted from a list at the program's website (www.biospice.org) established and maintained by SRI International, Menlo Park, CA, lead system integrator for Bio-SPICE, during the BioCOMP Program. The information has now moved to SourceForge.net because of completion of the DARPA program. In 2003, two special issues of OMICS, Volume 7, Numbers 3 and 4, were dedicated to Bio-SPICE with several articles highlighting contributions and technical progress made by various teams in the Bio-COMP Program. University of Texas research examined the usability of Bio-SPICE and several tools that work in the Bio-SPICE environment using their experience to create a tutorial manual for first-time users. Towards the end of the program CFD Research Corporation was funded to develop cellular modeling capability using partial differential equations. Researchers from Vanderbilt University were also brought in to examine how their expertise with model-based design and integration techniques could help facilitate the maturation of the tool integration framework. Naturally, the ease with which the tools worked in Bio-SPICE varied. Factors that influenced the performance of the tools were: funding levels of the different projects,

reliance on legacy programming, ability to adapt to change as the Bio-SPICE software environment evolved, computer skills of the research team, and experience of the team and the principal investigators.

4.0 Utilization of Computational Biology Tools

Prior to this effort there was limited hands-on experience within the Information Directorate with computational biology tools. The objective of this portion of the project was to examine the usability of the Bio-SPICE software environment and demonstrate how it could be used to develop biotechnology for computing. During this effort there were several releases of Bio-SPICE as it continued to evolve with a release happening every six months. Two college students employed for the summer by the Information Directorate at the Rome Research Site worked with Bio-SPICE. One of their assigned tasks required them to examine the usability of the current version of software during the time they were employed. The usability evaluation examined how easily the software could be installed, set up and used to run sample problems that tool developers provided with the software. Essentially the students, who were not computer science majors, but had biology backgrounds, were asked to use the documentation that was provided with the software to get it up and running. Since the students were each employed during two consecutive summers they naturally worked with different versions of Bio-SPICE. They worked with as many modules as possible until approximately six weeks were left in their summer term of employment. During the last six weeks the focus was shifted to demonstrating how Bio-SPICE could be applied to examining the role biology could play in computing. A summary of what the in-house team found while working with the Bio-SPICE will be presented starting with Bio-SPICE 4.0, then Bio-SPICE 6.0, and wrapping up with comments about initial work with Bio-SPICE 7.0.

Bio-SPICE 4.0

Bio-SPICE 4.0 represented a significant improvement in the Dashboard, the graphical user interface for Bio-SPICE, and the potential for linking different tools together graphically in a workflow. Figure 4 provides a snapshot of the Dashboard with a workflow on it. A workflow is the arrangement of some combination of input files, modelers, simulators, analyzers and output processes brought together by the user on the Dashboard. Improvement in documentation for the software was seen, but more refinement was still needed especially if someone with minimal computer skills was to try to use the tools. Ideally, a person with minimal computer skills would like to install the software and start using it right away, but with this version of the software it was found that more than minimal skills were needed to try to get the different software modules running. Numerous times pathnames had to be modified, installation scripts examined, and developers contacted to work the errors and bugs that were encountered. One demonstration involved linking the Datawarehouse, NCA Analyzer, Geneways, Graphviz, and Pathway Builder modules. Following the demonstration steps revealed one problem with documentation for evolving software. A module was created to link

Matlab with Bio-SPICE, that eliminated the requirement for using the NCA Analyzer, but the demonstration documentation was not updated to reflect this

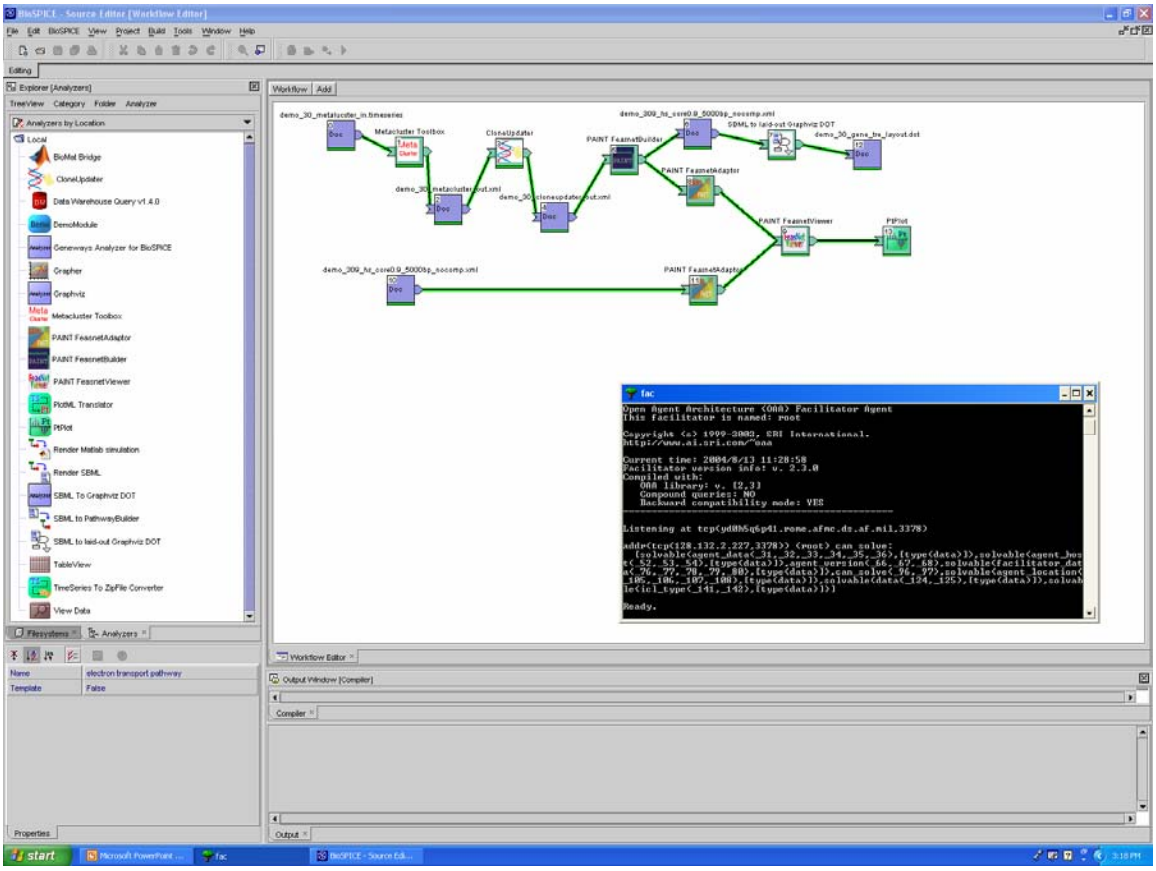


Figure 4. Snapshot of a Dashboard workflow.

change. All the modules were eventually installed, but another error was encountered when software available in Linux was not available in Windows. While a work around was available, it was determined that a significant amount of effort would be expended, preventing any examination of other tools. The decision was made to set aside this particular demonstration and look at other tools.

The next group of evaluated modules were: the Metacluster Toolbox, CloneUpdater, and the three PAINT modules: FeasnetBuilder, FeasnetAdaptor, and FeasnetViewer. The installation and building of the demo workflow are all outlined explicitly in a file. The installation of the software modules was pretty straightforward. FeasnetBuilder required a username and password to access the appropriate database. FeasnetViewer problems were encountered and after a few attempts to fix it the decision was made to move on to evaluating a different set of modules. The capability of this particular tool set, at the time of this evaluation, was found to be inadequate to demonstrate how Bio-SPICE could be used to develop biotechnology for computing.

During this time, a problem with the Bio-SPICE Dashboard occurred when trying to load saved workflows. The browsing window did not recognize workflow files as .wf extensions. The user had to change the dropdown menu to 'all files' and find the appropriate workflow. This feed back was provided to the Bio-SPICE developers.

BioCharon and BioCharon SketchPad were also installed and evaluated. SketchPad provides the user with the capability to draw out models of biological reactions. The graphical model is then analyzed using BioCharon, which converts it into the Charon language. These modules leverage University of Pennsylvania's previous work with Charon to now model and analyze hierarchical hybrid biochemical systems symbolically by state-space exploration techniques. The installation was just a matter of unpacking a zip file and executing the .jar file. To get the two programs to interact properly on the other hand was very unclear, requiring e-mail correspondence with a representative of the tools' developers to learn how to create a classpath from the command prompt to get the two programs to execute together and properly. The functionality of the program was checked by modeling a simple reaction, the electron transport chain which is the basis of aerobic respiration.[43]

The time had now arrived to shift the focus of the project to use the computational tools installed to implement a biocomputing technique or idea. G-protein biological switches, which are found throughout the body and used in adrenaline responses, were selected as the biological basis for the demonstration. It was envisioned that these switches could replace the logic gates that are the basis of silicon computers. A scheme for a binary adder was prepared using these G-protein switches instead of logic gates. The reactions that defined this biological switch were modeled using SketchPad and references to biology that were found on the internet. [20,21] Figure 5 shows the schematic of an electrical full adder circuit. Figure 6 illustrates a biological representation of the same circuit. Figure 7a shows a closer look at a protein switch used to create the Bioadder and its representation in Figure 7b.

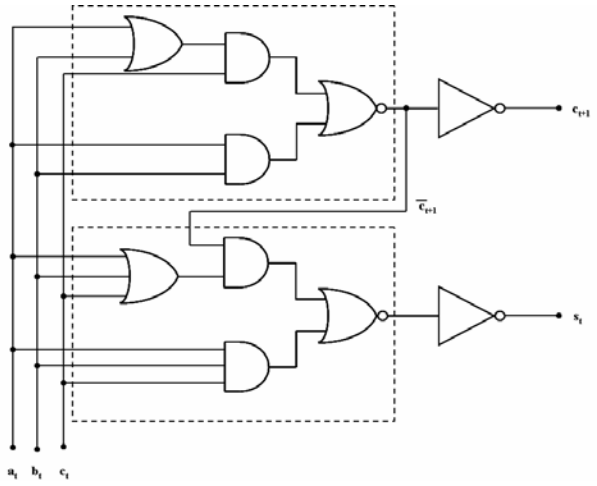


Figure 5. Illustration of a full adder.

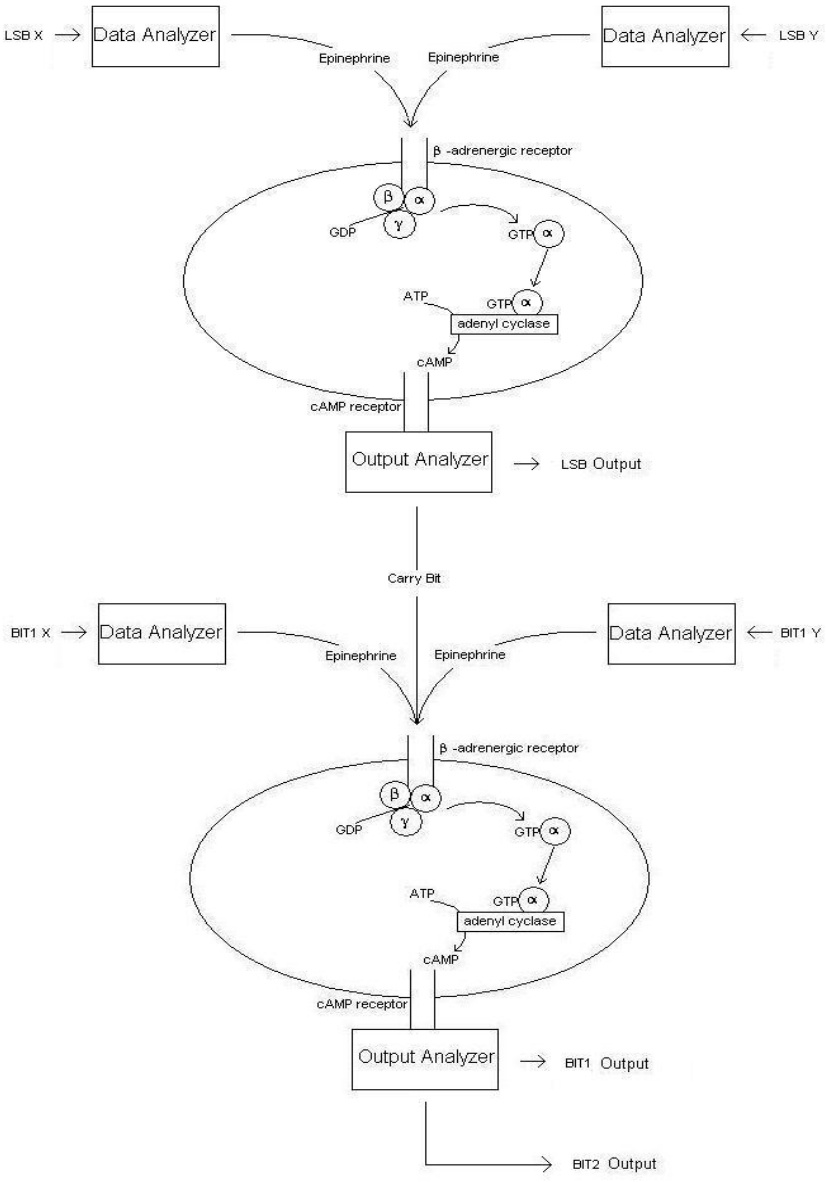
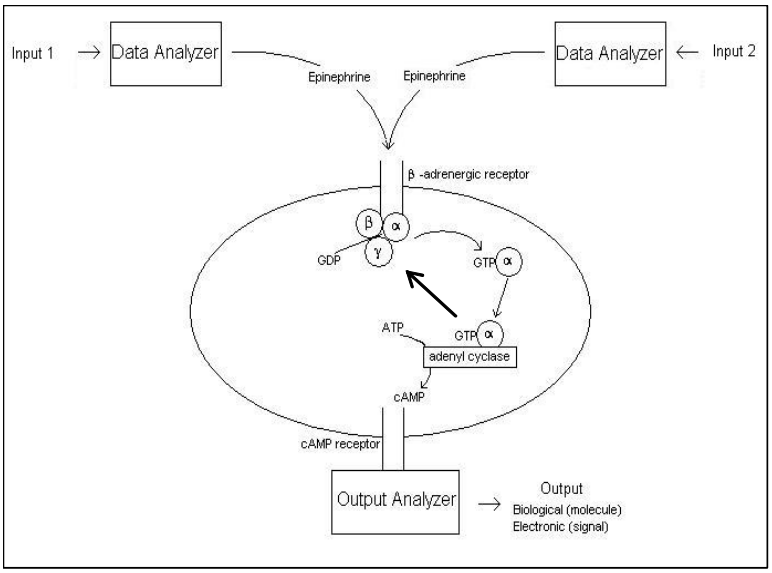
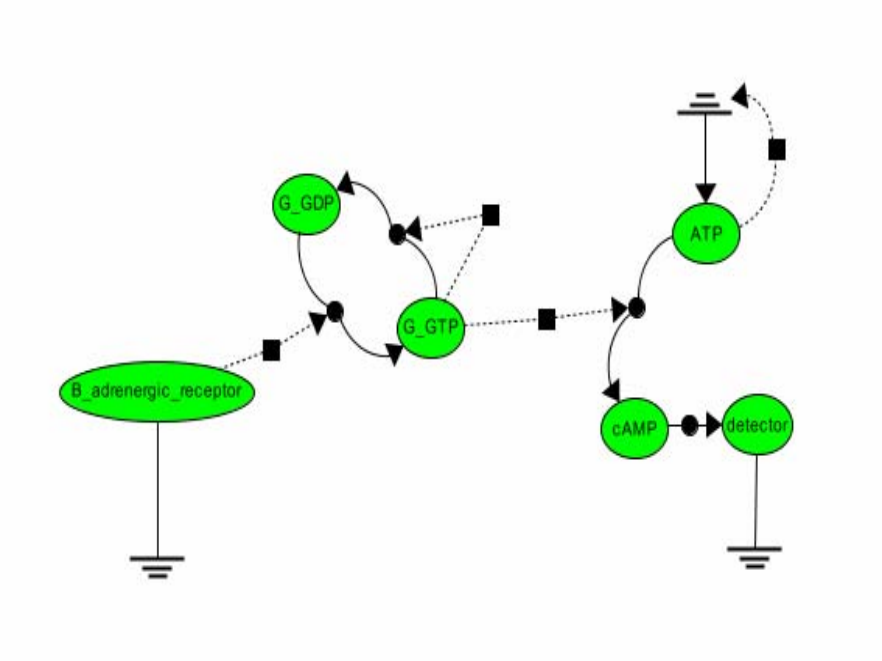


Figure 6. Illustration of a Bioadder.



(a)



(b)

Figure 7. Shows the concept of the protein switch (a) and how it is represented in Sketchpad (b).

Bio-SPICE 6.0

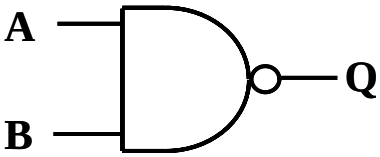
Since Bio-SPICE continued to evolve during the BioCOMP Program, another look at using Bio-SPICE and expanding upon the demonstration from the previous summer was warranted. A noticeable change to the software from past versions was the ability of clicking on a link for the update center which automatically installs the updates to various software modules. In past versions the user had to manually uninstall the software in order to load the latest versions including Bio-SPICE and its Dashboard. At times this seemed to be problematic since it was hard to determine if all of the components of the software, such as portions of the NetBeans development environment, were uninstalled completely. The process for doing manual updates was also improved because issues with legacy software made it difficult for some teams to make use of the automatic update. Better guidance was also provided to tool developers so that they could integrate their tools into Bio-SPICE. These improvements, however, didn't mean all tools were running better in Bio-SPICE. We found that tools that ran in the previous version did not run in the current version of the software. BioSketchPad, BioCharon, BioSpreadsheet, Exact Stochastic Simulator would run in stand alone configurations, but not in Bio-SPICE 6.0. The teams were contacted via e-mail and various options of installing the software and paths to software were examined, but the problem still existed. The problem was documented for the Bio-SPICE support team to address. Subtle variations in hardware and software configurations on individual platforms were blamed, but a definite answer was never pinpointed. Interestingly in this Bio-SPICE evaluation, JDesigner and Jarnac software, originally written for the Software Biology Workbench, not only ran in stand alone configuration, but also in the Bio-SPICE environment.

At the time this work was performed, JDesigner was found to be the most useful and robust model and simulation program for the development of the reaction pathways being examined. The tool allowed for a neat and accurate representation of the specific signal transduction pathway or gene regulation network. JDesigner contained several rate laws that could be integrated into the pathway or network and provides an option to customize a particular rate law in order to achieve a desired output. JDesigner provides three different modes of simulation and analysis: time course simulation, steady state analysis, and stoichiometry analysis. These three modes of simulation and analysis can produce either tabular or graphical output. When integrated with BioSPICE Dashboard v6.0, the analytical capability of the software was increased since several other operators can be applied to the given model and additional outputs obtained. JDesigner can convert the model designed in its environment into systems biology markup language (SBML) code. The versatility and user-friendliness of the JDesigner program led to its selection for the modeling of reaction pathways in this project.

After exploring the usability of Bio-SPICE, the focus shifted to expanding upon the demonstration performed the previous summer. The work explored a development path for a potential biomolecular computing concept which exploits proteins, gene network regulation and signal transduction pathways to execute computational functions. JDesigner provides an appropriate environment for the analysis of a known signal

transduction pathway or gene regulation network, and the design and development of an arrangement of computational components using a specific reaction pathway. A guanosine triphosphate (GTP) signaling pathway was constructed allowing the manipulation of different inputs in order to observe possible effects on the output of the system. An arrangement of computational components was also created consisting of a specific number of cascade cycles that functioned as NAND and NOT logic gates.[54] The ability of JDesigner to reproduce output that was obtained from experimental studies, supports its use as a tool for analysis and modification of known signal transduction pathways and gene regulation networks. The article also provided kinetic constants describing the rate laws for the stimulation and inhibition of adenylate cyclase in the presence of sodium chloride which was reproduced during the software evaluation.

The JDesigner program also provided an environment which facilitated the design of an arrangement of computational components using a reaction pathway to represent logic gates. A logic gate processes a signal that represents true of false.[18] The logic gate requires that the signal transduction pathway has a fast change in concentration of a particular protein or molecule, which is controlled by a threshold level on an input. The switch from one concentration level to another in a pathway is known as ultrasensitivity and it allows the implementation of Boolean functions to the reaction system [45]. The reaction pathway that was used to build a logic gate was the cyclic reaction between a kinase protein and a phosphatase protein, which is part of a signal transduction pathway. The ultrasensitivity property of this reaction pathway enables it to be manipulated to produce a logic gate. The steady-state analysis of the cyclic reaction yields a sigmoid curve and by decreasing the Km values of the enzymes, the reaction will increase the degree of sigmoidicity of the curve, resembling a switching action [45]. Using a cascade cycle a NOT gate and a NAND gate were created in JDesigner. Figure 8 illustrates a standard electronic NAND gate while Figure 9 shows the biological representation of the gate. The difference between the NOT gate and the NAND gate was that the cyclic reaction representing the NOT gate required one input, while the cyclic reaction representing the NAND gate required two inputs. These two gates were analyzed using the time course simulation to see what outputs would result from different inputs. The Vmax value and Km value of the Michaelis-Menten rate laws were manipulated to obtain the desired outputs for the NOT and NAND gate operators corresponding to specific inputs. These two logic gates were then used to produce an arrangement of computational components consisting of a NOR gate, an AND gate, and an OR gate as illustrated in Figure 10. Figure 11 shows the equivalent arrangement made up of NOT and NAND gates. This arrangement of computational components was analyzed using the time course simulation obtaining the desired output corresponding to specific inputs after manipulation of the Vmax value and Km value of the Michaelis-Menten rate laws inherent to the arrangement. Once again it was demonstrated that computational biology tools like those found in Bio-SPICE can be used to explore how biology can play a role in computing.



Input A	Input B	Input C
0	0	1
0	1	1
1	0	1
1	1	0

Traditional Symbol Truth Table

Figure 8. Illustration of a NAND gate with its truth table.

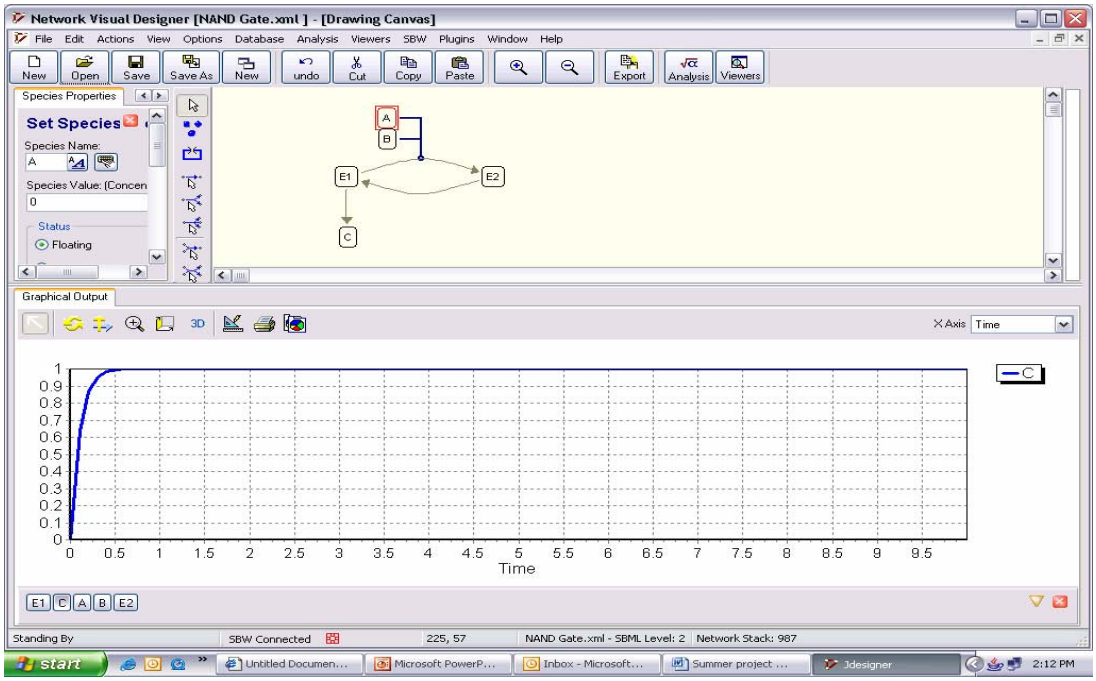


Figure 9. NAND gate in JDesigner.

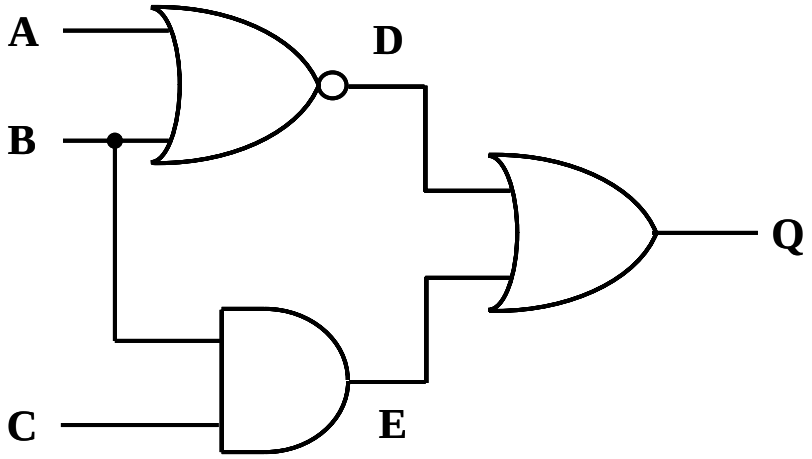


Figure 10. Combination of a NOR gate, an AND gate, and an OR gate.

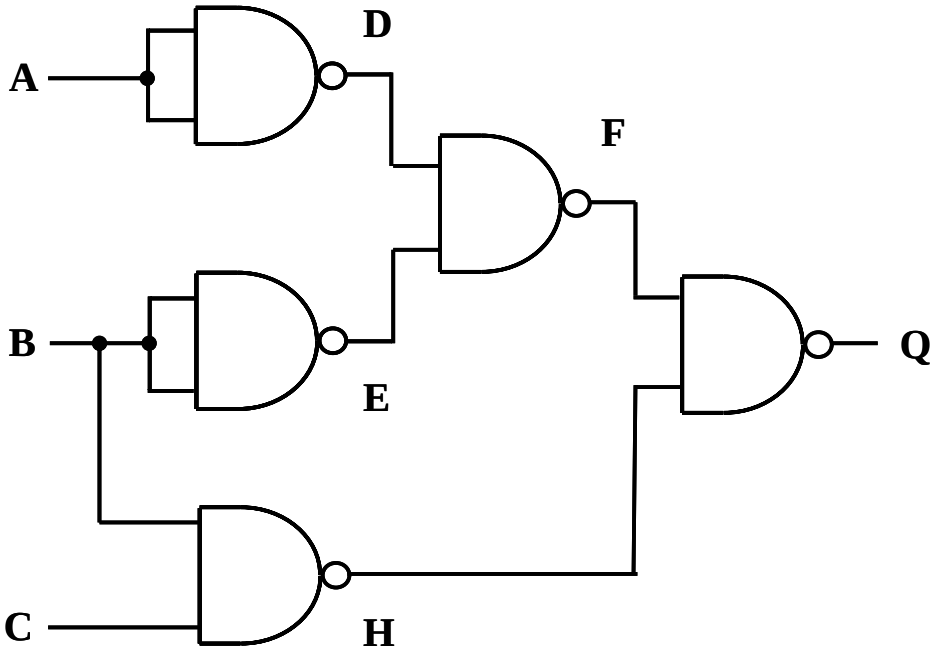


Figure 11. Another way to represent the function illustrated in Figure 10 using a combination of logic gates using only NOT and NAND gates

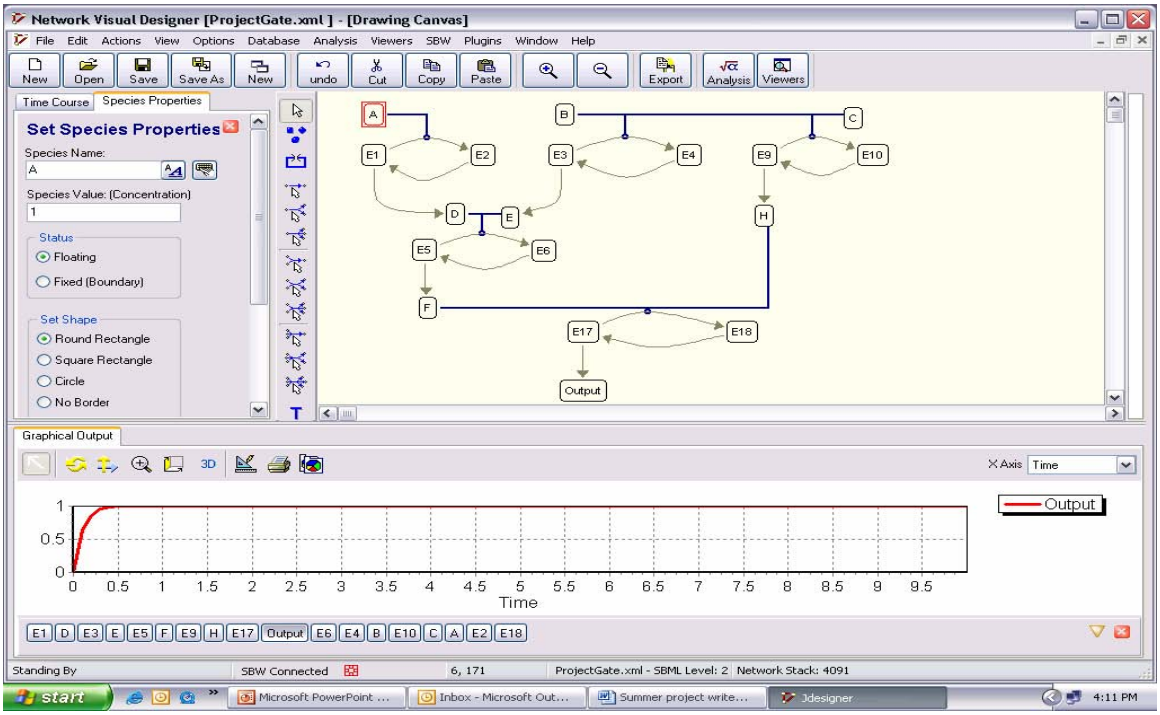


Figure 12. JDesigner representation of the logic gates shown in Figure 11.

Bio-SPICE 7.0

Evaluation of Bio-SPICE continued over time at different levels of intensity. Problems were brought to the attention of the appropriate individuals or teams. The in-house experience was valuable in providing feedback to the University of Texas team since they wrote a tutorial for first time users and SRI International because they were preparing the software for life after DARPA on SourceForge.net. As the BioCOMP Program winds down Bio-SPICE continues to be refined with more focus on getting better documentation and making the software more robust. Originally set up to run on Linux and Windows platforms, the software can now be run with Mac OS. The workflow is now portable across platforms, but now a new problem has arisen.

As the Air Force information assurance policies are being standardized, some issues have come to light that were not a problem with previous versions of Bio-SPICE. In the AFRL/IF computing environment, restrictions on those who have administrative privileges now impede researchers from trying to troubleshoot problems when working with an evolving piece of software. Fixes from tool developers under contract cannot be installed and exercised without an administrator. This introduces another barrier to learning and understanding the nature of a problem by reducing the opportunity to get hands-on experience. During evaluations of Bio-SPICE 4.0 and 6.0 several changes to the software were made each day for several days as attempts were made to get the software running correctly. Some software that was previously running will not run

under Bio-SPICE 7.0 because the user does not now have the correct permission. Permissions can only be changed by the administrator who has to be called back after solving one problem, then leaves to help others, only to be called back for the next problem which the user didn't know about because the previous problem prevented the software from running to the point of the next permission issue. Furthermore, Bio-SPICE was set up to be open source software which now requires special permission to install since it is non-standard software for the Air Force. There is currently no set time frame for granting or denying the use of open source software so the true impact on trying to use the software and meeting project milestones is not known. In the time of shrinking budgets the use of open source software, which has no purchase price or license fees, provides a cost effective means to explore how different software tools and techniques may or may not be used to develop new technologies. Commercial software can be expensive, especially when dealing with some of the electronic design automation tools used for computer hardware that would not be used every day or by a large number of researchers. Furthermore, it may only take twenty minutes of trying to use a tool to realize it will not be helpful. The potential exists for wasting resources on commercial software found to be inadequate for a research and development project if open source software cannot be readily used and evaluated. Significant manpower may be expended on the paperwork to install a non-standard piece of software only to find out after a few minutes that it is not adequate for the task at hand. Currently AFRL/IF is examining alternative arrangements with Air Force Information Assurance authorities to try and alleviate these impediments to efficient R&D software evaluation and development. No final decision has been made at this point on these alternatives.

5.0 Future Challenges for the Development of Computing Hybrids

In order for the concept of computing hybrids to be realized several more technology answers are needed to overcome the difficult task of integrating new science with proven technology. From the perspective of leveraging biotechnology one author stated that, "There is no longer any doubt as to the theoretical potential of DNA computing."[29] The challenge now is to develop a practical implementation of not only DNA computing, but other concepts determined to have strong potential.

If biology is to provide complementary computing technologies, the specific process needs to be understood. The appropriate models of the biological process targeted for exploitation must be created and validated. It was demonstrated that there are computational tools available that will help researchers understand biological processes for computing. There is room for improvement and more work will be needed. As the appropriate modeling and simulation tools and techniques are established for a particular process, the key information must be extracted and passed on to the next level of the design path. At this point it is not clear what that information is or in what form it must take. Those familiar with current state-of-the-art electronic design automation (EDA) would be comfortable if the new science could be implemented in the existing tools, but it is not clear if this is possible or if the tools are up to the task. Will a paradigm shift in the way a computer system is designed be needed to exploit biotechnology? It was stated

that, “Much work has already been done on shrinking the level of circuits and switches found in conventional silicon architecture down to the atomic level, using molecular wires, switches, and the Coulomb effect and ‘nano-islands’ to allow the encoding of a bit in the presence or absence of one electron...But we should remind ourselves again that this is not how biological organisms process information.” [48] Other researchers share this opinion and have begun looking at how to model the future systems, but we are far from a standardized process that is applicable to implementing a broad range of concepts. Furthermore, there is a need to establish the right metrics to gauge performance characteristics of the new computing concepts which also may require a shift from the current concepts. This gets even more complicated if one has to account for quantum effects along with other issues related to nanotechnology. Ideally it would be nice to have a software tool for the design of hybrid architectures that was an expert system that could address a mix of general purpose silicon computing technology, nanotechnology, biotechnology and quantum sciences. A user would sit down at a workstation, sketch a concept for a hybrid computing concept pulling in the components from a very large cell library, and let the software optimize the design. While much more work is needed before we will come close to the ideal capability, some research and development activity on a portion of this concept is already underway.[38] What is needed now is to select a couple of alternative computing concepts, and examine the process for building a hybrid architecture with them. The objective would be to determine how well the existing software tools and techniques can be utilized to identify the technology gaps against which to focus future research and development activity that would facilitate the maturation of the technology.

6.0 Concluding Remarks

This project provided the opportunity to gain hands on experience to complement the knowledge gained from participation in various DARPA Programs that involved biotechnology along with related modeling and simulation capabilities. It established a foundation on which new research projects can be built to take biotechnology from the laboratory into the field as advanced hybrid information systems. Much effort was spent exploring published research to understand what others have accomplished, problems they encountered, the potential of the biotechnology to advance computing technology, and to identify potential paths on which to focus future research. Various computational biology tools were also examined and found to have the potential to help develop biotechnology for future computing architectures. Key to the successful maturation of new technology is having the proper modeling and simulation tools and techniques to integrate the technology into future systems. Computational biology tools will play an important part in expanding the role biotechnology plays in computing. The next step is to select a few alternative computing concepts and examine how to integrate them into a hybrid architecture, defining a process for maturation of the technology for practical application.

7.0 References

1. Adleman, Leonard. 1994. Molecular Computation of Solutions to Combinatorial Problems. *Science*, Vol. 266, pp. 1021-1024.

2. Amorphous Computing HomePage. http://swiss.csail.mit.edu/projects/amorphous/#research. Link working as of August 14, 2006

3. Birge, Robert. 1997. *Protein-Based Branched-Photocycle Three-Dimensional Optical Memories.* Final Report RL-TR-96-274, April, DTIC Accession Number ADA327063.

4. Birge, Robert. 1997. *Branched-Photocycle Three-Dimensional Memory.* Final Report RL-TR-97-194, October, DTIC Accession Number ADA339787.

5. Birge, Robert, 2003. *Prototype Protein-Based Three-Dimensional Memory.* Final Report AFRL-IF-RS-TR-2002-319, January, DTIC Accession Number ADA412048.

6. Brenneman, Arwen, and Anne Condon. 2002. Strand Design for Biomolecular Computation. *Theoretical Computer Science*, 287, pp. 39-58.

7. Carroll, Steven. 2002 A Complete Programming Environment for DNA Computation. *Proceedings of the First Workshop on Non-Silicon Computing (NSC-1),* http://www.hpcaconf.org/hpca8/nsc.pdf, Held in Conjunction with 8th International Symposium on High-Performance Computer Architecture, Cambridge, MA, February 3, pp. 46-53.

8. Corne, David, and Gary Fogel. 2003. An Introduction to Bioinformatics for Computer Scientists. *Evolutionary Computation in Bioinformatics*. Gary B. Fogel and David W. Corne, Editors. Morgan Kaufmann Publishers/Elsevier Science, San Francisco, California. pp. 3-18.

9. Dasgupta, Dipankar, and Fablo Gonzalez. 2002. An Immunity-Based Technique to Characterize Intrusions in Computer Networks. *IEEE Transactions on Evolutionary Compuatation.* Vol. 6, No. 3, June. pp. 281-291.

10. Delcher, Arthur, Lee Hood, and Richard Karp. 1996. *Report on the DNA/Biomolecular Computing Workshop*, Supported by the National Science Foundation through Grant No. CCR-9642739 to the Johns Hopkins University, NSF 97-168, June 6-7.

11. Dwyer, Chris, Vijeta Johri, Moky Cheung, Jaidev Patwardham, and Alvin Lebeck. 2004. Design Tools for a DNA-guided Self-assembling Carbon Nanotube Technology. *Nanotechnology*, 15, pp.1240-1245.

12. Dwyer, Chris, John Poulton, Russell Taylor, and Leandra Vicci. 2004. DNA Self-assembled Parallel Computer Architectures. *Nanotechnology*,15, pp. 1688-1694.

13. Dwyer, Chris. 2005. A System-level Design Approach to the Evaluation of Self-assembled Computer Architectures. (Invited paper) *Proceedings of the 2nd Conference on the Foundations of Nanoscience: Self-Assembled Architectures and Devices*, April, pp. 255-261.

14. Frank, Michael. 2005. Approaching the Physical Limits of Computing. *ISMVL 2005, The Thirty-Fifth International Symposium on Multiple-Valued Logic.* IEEE Computer Society. May 19-21, University of Calgary, Calgary, Canada, pp. 168-185.

15. Garzon, Max, and Russell Deaton. 1999. Biomolecular Computing and Programming. *IEEE Transactions on Evolutionary Computation*, Vol. 3. No. 3, September. pp. 236-250.

16. Gillmor, Susan, Paul Rugheimer, and Max Lagally. 2002. Computation with DNA on Surfaces, *Surface Science*, 500, pp. 699-721.

17. Harner, Paul, Paul Williams, Gregg Gunsch, and Garl Lamont. 2002. An Artificial Immune System Architecture for Computer Security Applications. *IEEE Transactions on Evolutionary Computation*, Vol. 6, No. 3. June. Pp. 252-280.

18.Hewes, John. 2005. Logic Gates. The Electronics Club website at Kelsey Park Sports College. http://www.kpsec.freeuk.com/gates.htm. Link working as of August 14, 2006.

19. Hortos, William. 2003. An artificial immune system for securing mobile ad hoc networks against intrusion attacks. *Intelligent Computing: Theory and Applications*, Kevin L. Priddy, Peter J. Angeline, Editors, Proceedings of SPIE Vol. 5103. pp. 74-91.

20. http://web.mit.edu/esgbio/www/cb/membranes/gp.html. Part of the MIT Biology Hypertextbook. Link working as of August 11, 2006.

21. http://wilkes1.wilkes.edu/~terzaghi/BIO-226/lectures/16.html. As of August 11, 2006 this link was no longer working.

22. International Technology Roadmap for Semiconductors, 2005 Edition, http://www.itrs.net/Links/2005ITRS/Home2005.htm. Link working as of 28 July 2006.

23. Karp, Peter. 1994. *Report of the Workshop on Interconnection of Molecular Biology Databases.* SRI International Artificial Intelligence Center Technical Report SRI-AIC-549. Sponsored by the National Science Foundation and the Biomatrix Society in Stanford, California, August 9-12.

24. Kish, Laszlo. 2002. End of Moore's law: thermal (noise) death of integration in micro and nanoelectronics. *Physics Letters A*, 305, pp. 144-149.

25. Kumar, Srikanta, and Jordan Feidler. 2003. Editorial – BioSPICE, 2. *OMICS, A Journal of Integrative Biology*. Vol. 7, No. 4, p. 335.

26. Lesk, Arthur. 2002. *Introduction to Bioinformatics*. Oxford University Press Inc., New York.

27. Lewin, David. 2002. DNA Computing. *Computing in Science & Engineering*, May/June, pp. 5-8.

28. Livstone, Michael, Danny van Noort, and Laura F. Landweber. 2003. Molecular Computing Revisted: A Moore's Law. *Trends in Biotechnology*. Vol. 21, No. 3, March, pp. 98-101.

29. Maley, Carlo. 2003. DNA Computing and Its Frontiers. *Molecular Computing*, Tanya Sienko, Andrew Adamatzky, Nicholas Rambidi and Michael Conrad (Eds.), The MIT Press, Cambridge, MA, pp.153-189.

30. Mann, Charles. 2000. The End of Moore's Law? *Technology Review*. May/June. pp. 42-48.

31. Marcy, Duane, Bryan Vought and Robert Birge. 2003. Bioelectronics and Protein-Based Optical Memories and Processors. *Molecular Computing*, Tanya Sienko, Andrew Adamatzky, Nicholas Rambidi and Michael Conrad (Eds.), The MIT Press, Cambridge, MA, pp. 191-220.

32. McCaskill, John. 2001. Optically programming DNA computing in microflow reactors. *BioSystems*, 56, pp. 125-138.

33. McCaskill, John, Robert Penchovsky, Marlies Gohlke, Jorg Ackerman, and Thomas Rucker. 2001. Steady Flow Micro-Reactor Module for Pipelined DNA Computations Lecture Notes in Computer Science, Revised Papers from the 6th International Workshop on DNA-Based Computers: DNA Computing, June 13-17, 2000, Anne Condon, Grzegorz Rozenberg (Eds.) Springer-Verlag Vol. 2054, pp.263-270.

34. Meindl, James. 2003. Beyond Moore's Law: The Interconnect Era. *Computing in Science & Engineering*. January/February, pp. 20-24.

35. Paun, Gheorge. 2002. A Guide to Membrane Computing. *Theoretical Computer Science*, Vol. 287, pp. 73-100.

36. Paun, Gheorghe. 2004. Introduction to Membrane Computing, *First Brainstorming Workshop on Uncertainty in Membrane Computing*, Palma de Mallorca, Spain, November, pp. 17-65.

37. Paun, Gheorghe, and Radu Paun. 2005. *Membrane Computing as a Framework for Modeling Economic Processes.* http://psystems.disco.unimib.it/.

38. Pfeiffer, Anton, Tamal Mukherjee, and Steinar Hauan. 2003. Computer-Aided Synthesis of Microscale Electrophoretic Separation Systems in Confined Areas, *Proceedings of IMECE'03 2003 ASME International Mechanical Engineering Congress and Exposition*, November, pp. 16-21.

39. Reif, John. 1998. Paradigms for Biomolecular Computation. *Proceedings of the First International Conference on Unconventional Models of Computation* in Auckland, New Zealand, January, Unconventional Models of Computation edited by C. S. Claude, J. Casti and M. J. Dinneen, Springer Publishers, pp. 72-93.

40. Reif, John. 1998. Alternative Computational Models: A Comparison of Biomolecular and Quantum Computation. *Extended abstract of paper appeared in the 18th International Conference on Foundations of Software Technology and Theoretical Computer Science (FST&TCS98)*, December, pp. 102-121.

41. Reif, John. 2002. The Emerging Discipline of Biomolecular Computation in the US. *New Generation Computing*, Ohmsha, Ltd. and Springer-Verlag, 20, pp. 217-236.

42. Risch, L. 2002. The End of the CMOS Roadmap – New Landscape Beyond. *Materials Science and Engineering C*, 19, pp. 363-368.

43. Sasaki, Satoshi, and Isao Karube. 2003. Bioelectronics and BioComputers. *Molecular Computing*, edited by Tanya Sienko, Andrew Adamatzky, Nicholas Rambidi and Michael Conrad (Eds.), The MIT Press, Cambridge, MA, pp. 221-248

44. Salleh, Anna. 2006. DVD Uses Bug Protein to Store Data. *News in Science*, Australian Broadcasting Company, http://www.abc.net.au/news/stories/s1680304.htm . July 7.

45. Sauro, Herbert., and Boris Kholodenko. Quantitative analysis of Signaling networks. *Progress in Biophysics and Molecular Biology*, 86, 5-43 (2004).

46. Schaller, Bob. 1996. Moore's Law: The Benchmark of Progress in Semiconductor Electronics. http://mason.gmu.edu/~rschalle/moorelaw.html. September 26. pp. 1-28.

47. Sharma, Vijay. 2004. Is it Possible to Build Computers from Living Cells? *BioTeach Journal*, 2, 53-60.

48. Sienko, Tanya, Andrew Adamatzky, Nicholas Rambidi and Michael Conrad. 2003. Introduction: What is Molecular Computing? *Molecular Computing*, Tanya Sienko, Andrew Adamatzky, Nicholas Rambidi and Michael Conrad (Eds.), The MIT Press, Cambridge, MA, pp. xi-xvii.

49. Sienko, Tanya, and Jean-Marie Lehn. 2003. Molecular Recognition: Storage and Processing of Molecular Information. *Molecular Computing*, Tanya Sienko, Andrew

Adamatzky, Nicholas Rambidi and Michael Conrad (Eds.), The MIT Press, Cambridge, MA, pp. 33-61.

50. Smith, Warren. 1995. *Fundamental Physical Limits on Computation*. Technical Report, May 6. http://www.neci.nj.nec.com/~homepages/wds/fundphys.ps.

51. Tarankanov, Alexander O., Victor A. Skormin, and Svetlana P. Sokolova. 2003. *Immunocomputing:Principles and Applications*. Springer-Verlag, New York.

52. Theis, Thomas. 2003. Beyond the Silicon Transistor: Personal Observations. *Computing in Science & Engineering*. January/February, pp. 25-29.

53. Thiem, Clare, Steven Drager, Christopher Flynn, Thomas Renz, and Daniel Burns. 2004. Advanced Computer Technology for Novel Information Processing Paradigms. *Journal of Aerospace Computing, Information, and Communication.* July. Vol. 1. pp. 308-317.

54. Thomsen, William, and Richard Neubig. 1989. Rapid Kinetics of α2-Adrenergic Inhibition of Adenylate Cyclase. Evidence for a Distal Rate-Limiting Step. *Biochemistry*, 28, pp. 8778-8786.

55. Tuomi, Ilkka. 2002. The Lives and Death of Moore's Law. *First Monday*, Vol. 7. No. 11, November, pp. 1-35. http://www.firstmonday.dk/issues/issue7_11/tuomi/ .

56. Van Noort, Danny, Frank-Ulrich Gast and John S. McCaskill. 2002. DNA computing in Microreactors, *Lecture Notes in Computer Science:* Revised Papers. DNA Computing, 7th International Workshop on DNA-Based Computers, DNA7, Tampa, Florida, USA, June 10-13, 2001, Natasa Jonoska, Nadrian C. Seeman (Eds.), Springer. Vol. 2340, pp.33-45

57. Wagler, Patrick F., Uwe Tangen, Thomas Maeke, Steffen Chemnitz, Martina Juenger and John S. McCaskill. 2004. Molecular-Systems-on Chip (MSoC) Steps Forward for Programmable Biosystems. *Smart Strucutres and Materials 2004: Smart Electronics, MEMS, BioMEMS, and Nanotechnology*, edited by Vijay K. Varadan, Proceedings of SPIE Vol. 5389, SPIE, Bellingham, WA. Pp. 298-305.

58. Warren, Paul. 2002. The Future of Computing: New Architectures and New Technologies, Part 1: Biology Versus Silicon. *Engineering Science and Education Journal*. April. pp. 44-48.

59. Weiss, Ron. 2001. *Cellular Computation and Communications using Engineered Genetic Regulatory Networks*. Ph. D. dissertation, Massachusetts Institute of Technology.

60. Weiss, Ron, Subhayu Basu, Sara Hooshangi, Abigail Kalbach, David Karig, Rishabh Mehereja, and Ilka Netravali. 2003. Genetic Circuit Building Blocks for Cellular Computation, Communications and Signal Processing. *Natural Computing*, 2, pp. 47-84.

61. Weiss, Ron, George Homsy, and Thomas Knight, Jr. 1999. Toward in vivo Digital Circuits. *Dimacs Workshop on Evolution as Computation*. Princeton, NJ. pp. 1-18.

62. Wessner, Charles. 2003. Sustaining Moore's Law and the US Economy. *Computing in Science & Engineering*. January/February, pp. 30-38.

63. Zak, Michail. 2000. Physical Model of Immune Inspired Computing. *Information Sciences*, 129, pp.61-79.

Appendix A - Snapshot of Computational Biology Tools

Below is a snapshot of computational biology tools encountered while participating in DARPA's BioCOMP Program. The tools, like the Bio-SPICE environment itself continued to evolve.

Tool Name	Summary
BioGrid	BioGrid distributes simulation and analysis jobs across a cluster of computers by creating individual SBML files for each model parameter, simulating each model, and post-processing the output. On completion, BioGrid creates a web page with the results and links to relevant models and files created.
BioMat Bridge	BioMat Bridge integrates Matlab into the Bio-SPICE Dashboard, enabling a workflow to enter and exit Matlab's computation engine, permitting users to utilize any functionality supported by Matlab seamlessly within the Dashboard.
BioNetS	BioNetS simulates stochastic models of biochemical networks efficiently and accurately. Each chemical species can be specified as discrete or continuous, with discrete variables simulated with the Gillespie algorithm, and continuous variables simulated with chemical Langevin equations, or stochastic differential equations. The multiple simulation methods allows the use to incorporate the appropriate method.
BioSenS	BioSens provides a sensitivity analysis toolkit to investigate system output changes with respect to parameter variations. Sensitivity coefficients of the model are calculated using finite difference approximations, allowing the user to compute the system sensitivity to model parameters and initial conditions. In addition, Fisher Information Matrix based sensitivity measures provide a consolidation of the information in dynamical sensitivities and allow easy comparisons of the dependence of the overall system behavior with respect to the parameters.
BioSmokey	BioSmokey provides an easy well to manipulate time series data, with functions that include: column count, tab delimited conversion, column deletion, get column, get rows, row count, and sample.
BioSpreadsheet	BioSpreadsheet allows users to easily create new SBML models of biochemical processes or edit an existing SBML model.
BioWarehouse	BioWarehouse supplies a toolkit for constructing a warehouse of bioinformatics databases, including loaders for databases such as SWISS-PROT, KEGG, and NCBI Taxonomy.
Biowarehouse2SBML	Biowarehouse2SBML extracts stoichiometric and reversibility conditions for all reactions in a pathway in the BioWarehouse, producing an SBML model suitable for metabolic flux analysis.
CellX	CellX simulates multi-scale cell models while accounting for reaction and transport processes as well as the attendant intracellular gradients of composition. The simulations are performed on multiple dimensions, for example, reaction-transport equations are solved along fibrils (1-D), on membranes (2-D) and within bulk media (3-D), all simultaneously and with full coupling to accounts for molecular exchange between these domains.

Charon	Charon models and analyzes hierarchical hybrid biochemical systems symbolically by state-space exploration techniques. This symbolic analysis allows users to determine the feasibility of certain model behaviors, steady-state characteristics, etc.
Clone Updater	Clone Updater updates gene and EST clone annotations while identifying and eliminating redundant clones.
ESS	ESS performs stochastic simulations using an optimized version of Gillespie's exact stochastic simulation of coupled chemical species simulator.
Fluxor Computational Analyzer	Fluxor Computational Analyzer performs flux predictions resulting from specific reactions, selected by the user, having their fluxes limited or completely disabled.
Fluxor Spreadsheet	Fluxor Spreadsheet provides a spreadsheet-like interface to allow user to specify nutrient conditions, external metabolites, and gene knockouts, and to display the results of various kinds of flux predictions.
GeneCite	GeneCite permits users to specify sophisticated sets of queries and generates a table of the number of citations found for each query. The table can be presented as a web page or in standard spreadsheet format, allowing the user to view the full output of only those queries that generate an interesting number of citations.
GeneScreen	Genescreen processes gene expression data with a collection of computational statistic routines to extract significant gene association patterns.
Geneways	Geneways allows access to the Geneways Database, which contains literature from over fifty full text journals. Users can add additional information culled from literature to an existing biochemical model.
Homologue Finder	Homologue Finder searches for homologous genes or gene products for each gene or gene product in a pathway provided by the user. This results in a theoretical pathway in the target organism with found homologues.
Hybrid Automata Symbolic Reachability Tool	Hybrid performs parameter identification of biological systems through using piecewise affine hybrid automata.
JDesigner	JDesigner provides a visual biochemical network simulation tool that is part of the Systems Biology Workbench, permitting users to graphically specify the model, derive the representative set of differential equations automatically and generate a solution simulatable file (JDesigner itself has no simulation capability).
JigCell Tools (BioPack, Comparator, Model Builder, Project Manager, Run Manager)	JigCell Tools includes several capabilities in one package. The Model Builder builds and edits SBML models using a spreadsheet interface, reducing models by finding conservation relations in the equations. The Run Manager defines an ensemble of runs that specify how to simulate SBML models with various changes to parameter and initial condition values. XPP provides access to Bard Ermentrout's XPP simulator. Biopack provides access to the numerical integration routines of LSODAR. The Comparator executes an ensemble of runs, applies data transformations to the simulation output, and determines how well the model matches the collected experimental results it is attempting to reproduce. The Parameter Estimator fits model parameters by applying local and global optimizers to find values that bring the model and experimental results into close agreement. Compare2 allows the user to visualize the change in the fit of a model to

	experimental results as a model is developed over time. The Project Manager stores and organizes data files used by other tools into logically related projects.
Karyote Cell Analyzer/KCA	KCA provides a spatial whole-cell or multi-cell modeling capability. Taking into account a network of biochemical processes, intracellular architecture, membrane transport properties, initial cell state and conditions in the extracellular system, Karyote Cell Analyzer simulates the time course of all chemical species in all cellular compartments.
Karyote Genome Analyzer/KAGAN	KAGAN generates detailed biochemical information about gene control networks by analyzing raw time series microarray data. KAGAN can also use timeseries data integrated with a subset of a cell biomic simulator to perform cell simulations despite model incompleteness.
lcDNA	lcDNA estimates confidence intervals for mRNA expression levels in microarray experiments, including elimination of extreme outliers, quality filtering, normalization of the log10 signal intensity ratios, and assessment of expression levels.
MetaCluster	MetaCluster performs cluster analysis with several algorithms, including the metacluster algorithm, which identifies trends insensitive to the clustering method used, implying a higher confidence in the clustering analysis results. MetaTool derives conclusions about the pathway structure of a metabolic network from the stochiometric reaction equations and information about reversibility and irreversibility of enzymes.
MIAMESpice	MIAME Spice packages raw and normalized datafiles from a set of related microarray experiments, saving all associated data from an experiment (or set of experiments) into one archive file. Users can also enter experimental annotations, array design information, and array design files.
Monod	Monod supports collaborative work through a web-based environment, with tools for text-based data mining, note annotation, literature annotation, user workspace with selective sharing properties, and model editor.
NCA	NCA uses available connectivity information between genes and transcriptional factors and gene expression level time course data (obtainable through DNA microarray experiments) to estimate parameters and infer a gene transcriptional network through a Matlab analysis routine.
NYUMAD	NYUMAD facilitates storage of microarray data and simulation time series data in a relational data base management system, including front-end capabilities for data presentation and maintenance.
NYUSIM	The NYU Simulation Database (NYUSIM), a subset of NYUMAD, stores time series data from simulations, allowing manipulation and cataloging of these traces.
OctaveBridge	OctaveBridge provides an interface to a free scripting language and simulation tool that can be used as an alternative to Matlab for time series data.
PAINT	PAINT allows researchers to visualize, identify, and analyze gene regulatory regions, and test for statistical significance to rank the likelihood of the involvement of individual transcription factors in the biology under study. The results are written in suitable formats for further analysis using modeling and simulation tools.

Pathway Screen	Pathway Screen generates a the pathways that a list of genes are incorporated in and a link to a picture of these pathways.
PathwayBuilder	Pathway Builder provides software tools that can be used to graphically represent a biological pathway, formally define conceptual models of processes in the pathway, include mathematical representations of those models, and produce representations of these models executable in Matlab or SBML compliant simulators.
SAL	SAL allows highly automated abstract modeling of biological pathway networks to accelerate understanding. SAL uses modern automated symbolic logic to represent biological systems, allowing biologists to understand complex biological systems, make hypotheses, and refine models.
SBW Optimizer	SBW Optimizer provides an interface between Matlab and the Systems Biology Workbench (SBW), allowing users to access existing SBW modules or create new modules.
Sensitivity Analyzer	Sensitivity Analyzer performs a response surface analysis of a biochemical model. The user selects features of the model behavior and the tool performs an efficient simulation of the system under different parameter assumptions to better understand the dependence of the selected feature values on the parameters.
Simpathica	Simpathica simulates a biochemical model with differential algebraic description, or ordinary differential equations. The model is entered using a library of template reactions. Simpathica also offers an analysis backend based on a temporal logic model checker capable of analyzing several traces at a time. The analyzer provides feedback to the user about particular events observed in the behavior of the system.
SOS Tools	SOS Tools solves feasibility or optimization sum of squares problems with a third-party Matlab toolbox. The solution is arrived at by performing sum of squares decomposition for multivariate polynomials, which is efficiently computed with semi-definite programming.
Strongly Connected Component Finder SCC/MotifAnalyzer	SCC searches for graphical motifs conserved across pathways to better understand feedback. The program searches for strongly connected components, two components for which each component is found to be strongly connected to the other, indicating that the two components are very likely to belong to a common function and control structure.